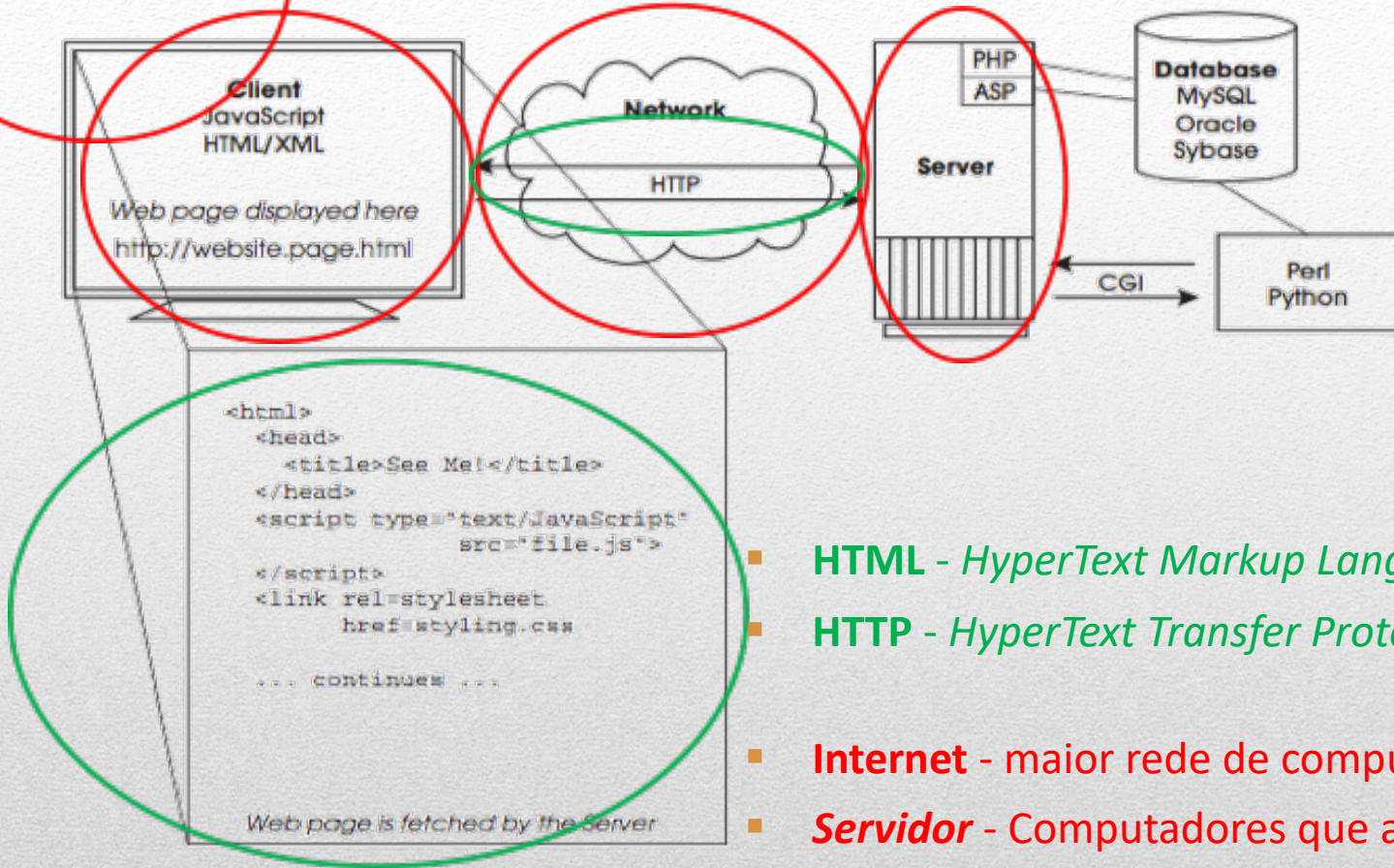




# **Design de Websites Responsivos**

---

## World Wide Web



- **HTML** - *HyperText Markup Language*
- **HTTP** - *HyperText Transfer Protocol*
- **Internet** - maior rede de computadores mundial
- **Servidor** - Computadores que alojam os ficheiros da Web
- **Browser** - programa que corre no computador para mostrar os ficheiros da Web



## Web: Documentos HTML



- World Wide Web coleção de documentos HTML
- Documentos ligados através de Hyperlinks

## HTML

- ***HyperText Markup Language - HTML***
    - Linguagem de etiquetas (*tags*) utilizada para produzir páginas na Web
    - Documentos HTML podem ser interpretados por *browsers*
    - Tim Berners-Lee criou o HTML original em 1993
-

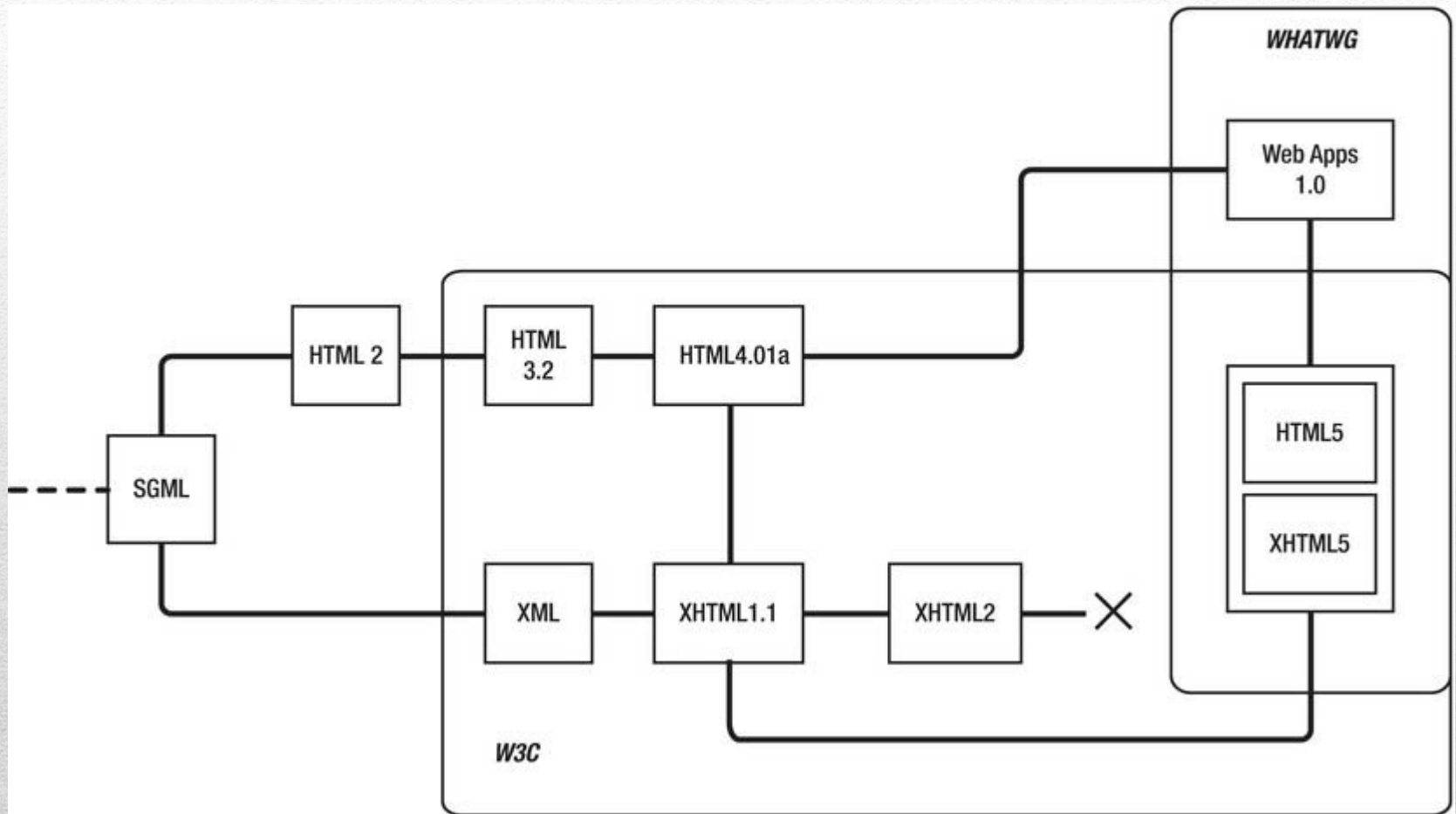


## HTML - História

### ■ História

- **1993** - início do HTML com Tim Berners-Lee
  - **1994** - formado o W3C (World Wide Web Consortium)
  - **1995** - HTML 2.0, primeira especificação oficial pelo IETF
  - **1996** - versões do HTML passam a ser mantidas pelo W3C
  - **1997** - definida a especificação do HTML 3.2
  - **1998** - W3C propõe o HTML 4.0
  - **1999** - definida a especificação do HTML 4.01
    - | - durante este período o W3C dedica-se ao XHTML
  - **2004** - WHATWG começa a trabalhar na especificação “Aplicações Web” (início do HTML5)
  - **2006** - W3C começa a trabalhar no HTML5
  - **2009** - fim do XHTML 2.0, maior concentração no HTML5
  - **2012** - 1ª recomendação oficial do HTML5 pelo W3C
-

## Evolução do HTML





## HTML - Tags

Etiqueta	Descrição
<html>...</html>	Delimita o início e o fim do documento
<head>...</head>	Delimita a zona de cabeçalhos
<title>...</title>	Delimita o título
<body>...</body>	Delimita o corpo do documento
<h1>...</h1>	Delimita uma secção de nível n 0-6
<ul>...</ul>	Delimita uma lista não numerada
<ol>...</ol>	Delimita uma lista ordenada
<li>...</li>	Delimita um item da lista
 	Mudança de linha
<hr />	Insere uma linha horizontal
<p>...</p>	Delimita um parágrafo
	Insere a imagem indicada no atributo src
<a href="...">...</a>	Delimita um hyperlink para a página indicada em href
<div> ... </div>	Define um grupo de elementos (Divisão)
<span> ... </span>	Define um grupo de elementos (inline)
<!--...-->	Delimita uma zona de comentários

- <http://www.w3schools.com/>

## HTML – *Tags* de Elemento Único

Elementos	Descrição
<area />	Define uma área numa imagem-mapa
<base />	Define URL base para todos os URLs relativos no documento
 	Mudança de linha
<col />	Define propriedades para cada coluna num <colgroup>
<embed />	Define um contentor para uma aplicação externa (e.g.,Flash)
<hr />	Insere uma linha horizontal
<img />	Insere a imagem indicada no atributo src
<input />	Define um campo entrada para o utilizador inserir dados
<keygen />	Especifica um campo gerador de chaves (keys)
<link />	Define uma ligação entre o documento e um recurso externo
<meta />	Permite colocar meta-informação não visível sobre o documento
<param />	Define parâmetros para <i>plugins</i> embebidos num elemento <object>
<source />	Especifica vários recursos <i>media</i> em elementos <i>media</i> (e.g., <audio> ou <video>)
<track />	Define zonas de texto para elementos <i>media</i> (e.g., <audio> ou <video>)
<wbr />	Especifica onde colocar num texto uma mudança linha (e.g., num elemento <p>)

- HTML5 o carácter “/” pode ser omitido
-



## CSS

### ■ **CSS – *Cascading Style Sheets***

- Descrição de regras de estilo que indicam ao *browser* como apresentar os diversos elementos de um documento HTML

### ■ **Motivação**

- HTML 3.2 - necessário formatar cada página individualmente
  - Separação entre as regras de apresentação e os conteúdos permite utilizar as regras de estilo em várias páginas
-

## CSS – Sintaxe

### ■ Regras de formatação - Sintaxe

- Selector
- Bloco de declarações (define propriedades)

```
selector [, selector2, ...] [:pseudo-class] {  
    property: value;  
    [property2: value2; ...]  
}
```

- Exemplos

```
h1{  
    color: red;  
    text-align: left;  
    font-size: 8pt;  
}
```

---



## CSS – Selector

<b>.class</b>	Selecciona os elementos com class=“nomeClasse”	.nomeClasse
<b>#id</b>	Selecciona o elemento com o id=“identificadorX”	#identificadorX
<b>elemento</b>	Selecciona todos os elementos desse tipo	h1
<b>*</b>	Selecciona todos os elementos	*
<b>Elem1, Elem2</b>	Selecciona os elementos Elem1 e Elem2	h1, h2
<b>[atributo]</b>	Selecciona os elementos que tenham o atributo	[href]
<b>Elem1&gt;Elem2</b>	Selecciona todos Elem2 que sejam filhos de Elem1	Pessoa>Nome

## HTML e CSS

- **Formas de inserir folhas de estilo em HTML**

- 1 - Folha de estilo externa

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```
</head>
```

- Estilo aplicado em várias páginas
      - Formatação é modificada alterando apenas um ficheiro
      - Documentos incluem a folha de estilo utilizando a *tag* `<link>`
-



## CSS – *mystyle.css*

```
h1{
    color: red;
    text-align: left;
    font-size: 8pt;
}

a:visited{
    color: blue;
}

#para1{
    text-align: center;
    color: black;
}

.center{
    text-align: right;
}
```

---

## HTML e CSS: Exemplo

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css"
href="css/mystyle.css">
</head>

<body>
    <p id="para1">Hello World!</p>
    <p>This paragraph is not affected by the style.</p>
    <h1>This paragraph has a different format</h1>

    <p class="center"> Center-aligned heading</p>
</body>
</html>
```

---



## HTML e CSS: Folha de Estilo Interna

- 2 - Folha de estilo interna

```
<head>

<style>
  h1{
    text-align: right;
    font-size: 20pt;
  }

  p {margin-left: 20px;}
</style>

</head>
```

- Documento tem um estilo único
  - Definidos na secção `<head>` utilizando a *tag* `<style>`
-

## HTML e CSS: *Inline Styles*

- 3 - Formatação *inline* (*inline styles*)

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

- Estilo é aplicado a uma única ocorrência de um elemento
- Utilizado o atributo `style` da etiqueta do elemento





## HTML e CSS: Várias Folhas Estilos

- **Folha de estilo externa**

```
h1{  
    color: red;  
    text-align: left;  
    font-size: 8pt;  
}
```

- **Estilo (interno) definido na página**

```
h1{  
    text-align: right;  
    font-size: 20pt;  
}
```

- **Inserindo a folha de estilo externa na página**

```
h1{  
    color: red;  
    text-align: right;  
    font-size: 20pt;  
}
```

---

## HTML e CSS: Prioridades

### ■ Várias folhas de estilo

- Folha de estilo resultante é a conjunção de todos as propriedade definidas
- Folhas de estilo prioritárias fazem *override* as propriedades definidas para o mesmo seletor

### ■ Prioridades

- |                               |                    |
|-------------------------------|--------------------|
| ■ 1 - <i>Browser</i>          | (menor prioridade) |
| ■ 2 - Folha de estilo externa |                    |
| ■ 3 - Folha de estilo Interna |                    |
| ■ 4 - Estilo <i>inline</i>    | (maior prioridade) |



# HTML5

---

## O que é o HTML5?

- Versão mais recente e mais abrangente do HTML
  - Desenvolvida pelo grupo WHATWG e depois normalizada pelo W3C para desenvolver **Aplicações para a Web**
  - Nova versão do HTML4, XHTML1 e DOM nível 2
-



## Qual a diferença?

- API's em Javascript para novas funcionalidades
  - Novas *tags*
  - Novos atributos
-

## HTML5?

- Compatível com as versões anteriores ao contrário do XHTML 2.0
  - Especifica a forma como os erros devem ser geridos
  - Garante a compatibilidade entre dispositivos e *browsers*
  - Reduz a necessidade de *plug-ins* (ex: Flash e Silverlight)
-



## HTML5: Composição

- Novos elementos de estrutura da página
    - *section, header, hgroup, footer, nav, article e aside*
  - Novos elementos para o conteúdo
    - *figure, figcaption, mark e time*
  - Web Forms 2.0 (formulários)
  - Novos elementos Multimédia
    - *audio, video e canvas 2D/3D*
  - Microdata
  - ...
-

## HTML5: API

- **API incluídas pelo W3C**
    - CSS3
    - Web Fonts
    - Geolocation
    - SVG (Scalable Vector Graphics)
    - Web Storage
    - Web Workers
    - WebSockets
    - Offline Applications
    - Drag and Drop
-



## HTML5: Início do Documento

- **DOCTYPE e Tipo de Caracteres**
  - **HTML 4.01**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.  
dtd">
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=utf-8">
```

- **HTML5**

```
<!DOCTYPE html>
```

```
<meta charset="utf-8">
```

---



## HTML5: Metadados

```
<!DOCTYPE html>  
<html>
```

```
<head>
```

```
<title>Title of the document</title>
```

```
<meta name="description" content="Free Web  
tutorials">
```

```
<meta name="keywords"  
content="HTML,CSS,XML,JavaScript">
```

```
<meta name="author" content="Rui Jesus">
```

```
<meta http-equiv="refresh" content="30">
```

**<meta>**  
HTML5 acrescenta um novo  
atributo  
"charset"

```
<meta charset="UTF-8">
```

```
<link rel="stylesheet" type="text/css"  
href="css/mystyle.css">
```

```
</head>
```

**<meta>**  
Descreve metadados num  
documento HTML

Link para o documento CSS



## HTML5: Tags de Multimédia e Formulários

<i>Tags</i>	Descrição
<canvas>	Etiqueta que recorre à utilização de <i>scripts</i> para criar animações e desenhar gráficos vetoriais
<audio>	Define conteúdo relativo a áudio digital
<video>	Define conteúdo relativo a vídeo digital podendo conter áudio digital
<source>	Permite definir vários media para os elementos <video> e <áudio>
<embed>	Define um ponto de ligação para uma aplicação externa (e.g., FLASH)
<track>	Define <i>tracks</i> de texto para elementos media

<i>Tags</i>	Descrição
<datalist>	Define uma lista pre-definida de opções para controlos de <i>input</i>
<keygen>	Permite gerar um par chave-campo
<output>	Define o resultado de um cálculo

---

## HTML5: Elementos Semânticos/Estruturais

<i>Tags</i>	<b>Descrição</b>
<article>	Representa um conteúdo específico dentro de um documento. Pode ser reutilizado no mesmo documento
<aside>	Representa a secção de uma página cujo conteúdo é tangencialmente relacionado com o conteúdo que o rodeia
<section>	Define uma secção que é um conjunto em que estão agrupados elementos sobre um tema comum
<figure>	Elemento que pode ser usado para agrupar elementos visuais
<figcaption>	Permite associar texto para um elemento <figure>
<footer>	Define um <i>footer</i> para um documento ou secção
<header>	Define um <i>header</i> para um documento ou secção
<hgroup>	Agrupa um conjunto de elementos <h1> a <h6> quando <i>heading</i> tem vários níveis
<mark>	Permite marcar texto ( <i>highlighted text</i> )
<time>	Define a data/hora
<nav>	Representa uma secção da página que contém <i>links</i>

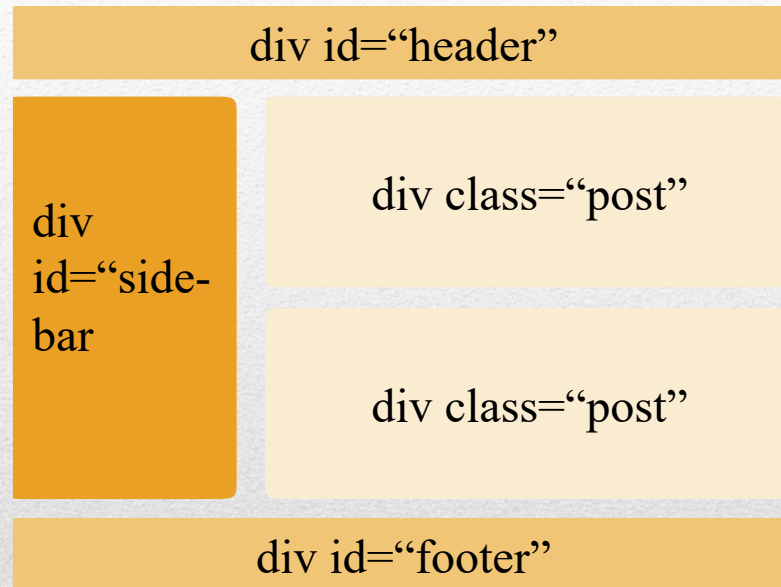


## HTML5: *Tags* Removidas

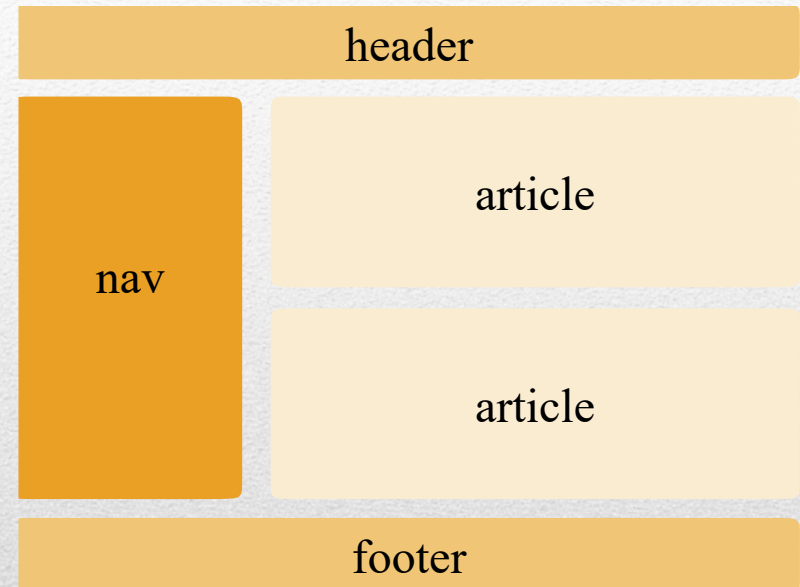
- ***Tags* do HTML 4 não suportadas no HTML5**
    - <acronym>
    - <applet>
    - <basefont>
    - <big>
    - <center>
    - <dir>
    - <font>
    - <frame>
    - <frameset>
    - <noframes>
    - <strike>
    - <tt>
-

## HTML5: *Tags* de Estrutura (I)

### HTML 4



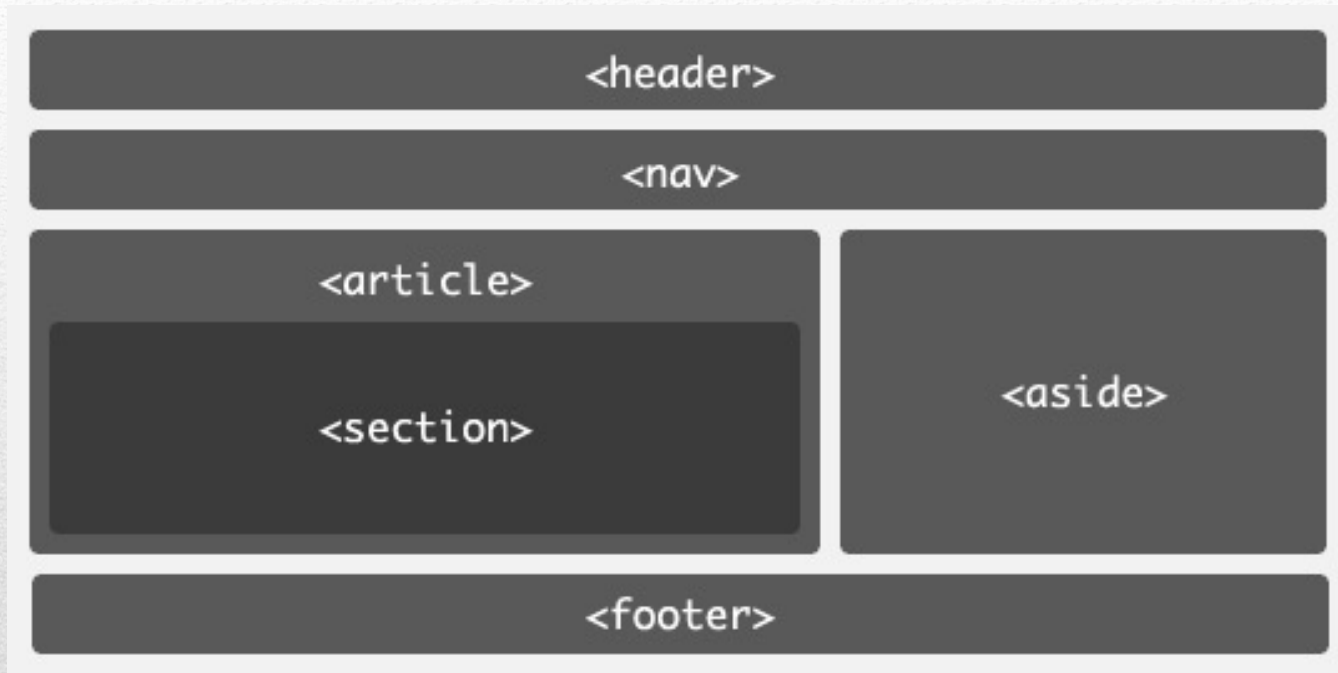
### HTML5



- HTML4 - *tag* `<div>` era utilizada para definir todo o tipo de secção
  - HTML5 - novas *tags* apropriadas para as várias secções
-



## HTML5: *Tags* de Estrutura (II)



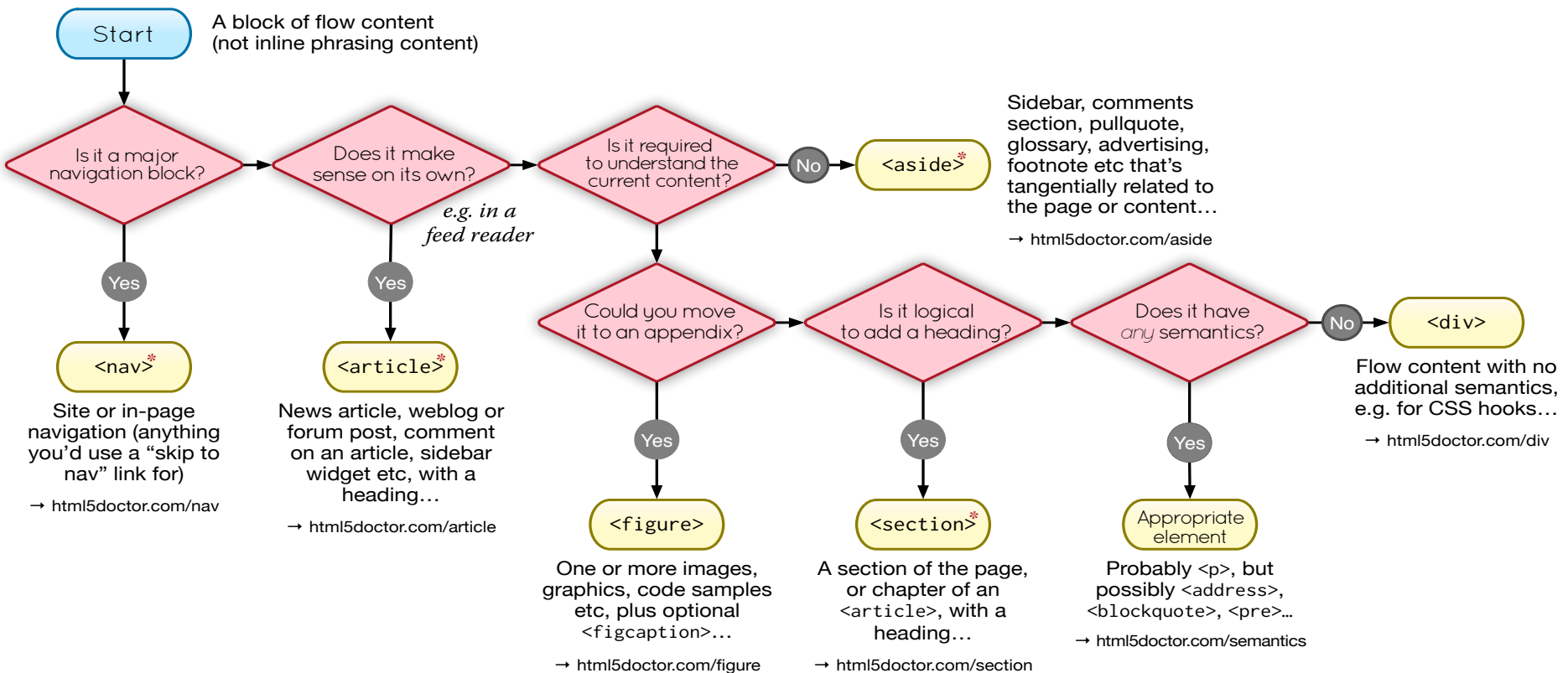
- **<section>** para agrupar temas relacionados pelo conteúdo
  - **<article>** para conteúdo independente, que não precisa de mais nada para ser entendido
-

## HTML5: Tags de Estrutura (III)



### HTML5 Element Flowchart Sectioning content elements and friends

By @riddle & @boblet  
[www.html5doctor.com](http://www.html5doctor.com)



\*Sectioning content element

These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline  
→ [html5doctor.com/outline](http://html5doctor.com/outline)



## HTML5: Exemplo (I)

**What is HTML?**

HTML is the acronym for HyperText Markup Language

**Annotations:**

- CSS3 rotate transform**: Points to the 'Multimedia Authoring' text in the top-left corner.
- <header> elements**: Points to the main title and subtitle area.
- <section> element**: Points to the 'Unordered List Tag' section header.
- CSS3 rounded borders**: Points to the rounded corners of the 'Unordered List Tag' content box.
- <nav> elements**: Points to the 'Table Tags' link in the sidebar navigation menu.
- container <div> element**: Points to the sidebar area containing the navigation menu.
- <aside> element**: Points to the 'HTML Lesson' section in the right sidebar.
- <footer> elements**: Points to the 'Color Codes' section at the bottom of the main content area.

**Page Content:**

**Unordered List**

**Ordered List**

**Color Codes**

**Table Tags**

**Unordered List Tag**

The language is made up of a set of codes and symbols that will generate:

- Text,
- Images,
- Sounds,
- Frames
- Animation on your browser pages.

**Ordered List Tag**

Two common ways to start your website are either:

1. To create your own codes in a simple text editor and save them as a .html file.
2. Use a HTML generator software.

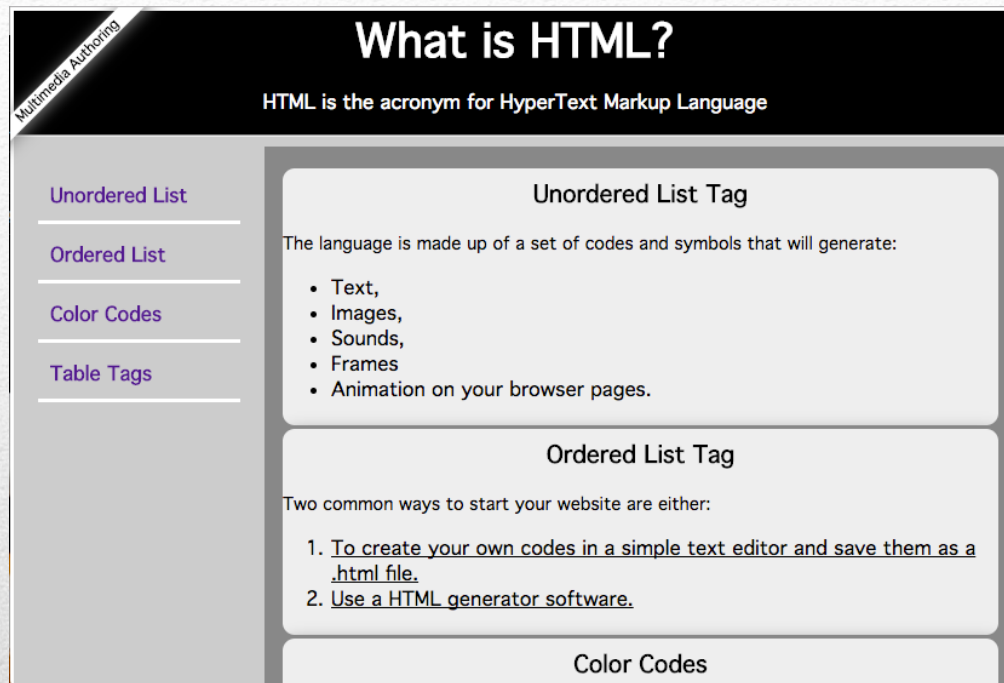
**Color Codes**

This section will give you information about color tags

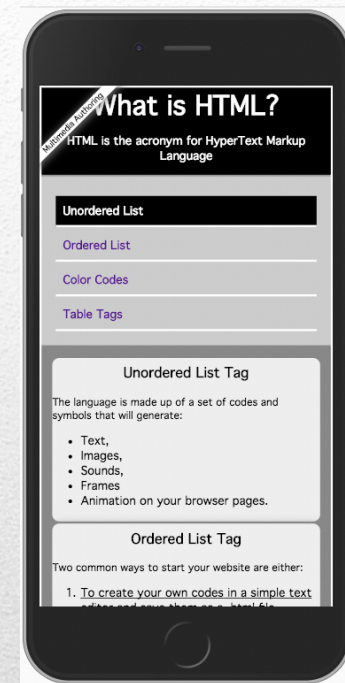
**HTML Lesson**

*The aside tag defines some content aside from the content it is placed in.*

## HTML5: Exemplo (II)



Tablet



Smartphone

- Boas práticas recomendam que os Designers devem começar a desenvolver a versão para Smartphone



## HTML5 : Exemplo III

```
<header>  
  <h1>Header</h1>  
  <h2>Subtitle</h2>  
</header>
```

```
<nav>  
  <h3>Nav</h3>  
</nav>
```

```
<div id="container">  
  <section>  
    <p>Section 1</p>  
  </section>  
  <section>  
    <p>Section 2</p>  
  </section>  
  <section>  
    <p>Section 3</p>  
  </section>  
</div>
```

```
<footer>  
  <h2>Footer</h2>  
</footer>
```

Tags de Estrutura  
Ficheiro  
"slideshow\_1.html"





## HTML5: Exemplo (IV)

```
body {
  background-color: #CCCCCC;
  font-family: Geneva, Arial, Helvetica, sans-serif; margin: 0px;
  max-width: 100%;
  border: solid;
  border-color: #FFFFFF;
}
header {
  background-color: #000000;
  display: block;
  color: #FFFFFF;
  text-align: center;
}
h1 {
  font-size: 2.441rem; margin: 0px;
}

section {
  display: block;
  background-color: #eee;
  margin: 3px auto;
  padding: 3px 0px;
  border-radius: 10px;
  box-shadow: 2px 2px 20px #888;
}

section h2 {
  border-radius: 10px;
  padding: 5px;
  font-size: 1.2rem;
}
```

---



## CSS3

Propriedades	Descrição
<code>border-radius</code>	Atribuir cantos arredondados a qualquer elemento
<code>border-image</code>	Permite usar uma imagem como margem ( <i>border</i> ) de um elemento
<code>background-image</code>	Permite usar mais do que uma imagem como fundo de um elemento
<code>linear-gradients</code>	Permite mostrar transições suaves entre duas ou mais cores
<code>box-shadow</code>	Atribuir sombra a um elemento
<code>@font-face</code>	Permite usar novas fontes de texto que não estão instaladas
<b>transform 2D 3D</b>	Conjunto de propriedades que permitem realizar transformações 2D e 3D a um elemento
<code>transitions</code>	Permite criar efeitos de transição, alterando suavemente o valor de propriedades
<b>animations</b>	Conjunto de propriedades para definir animações a elementos
<b>multiple columns</b>	Conjunto de propriedade que permitem definir várias colunas de texto
<code>box-sizing</code>	Permite incluir <i>padding</i> e <i>border</i> no valor total de <i>width</i> e <i>height</i>

---

## CSS3 Animations

```
.animation {  
    width: 20px;  
    height: 20px;  
    background-color: red;  
    position: relative;  
    animation-name: example; animation-duration: 4s;  
    animation-iteration-count: infinite;  
    animation-timing-function: linear;  
    border-radius: 10px;  
}  
@keyframes example {  
    0% {background-color:white; left:0%; top:0%;}  
    25% {background-color:black; left:50%; top:50%;}  
    50% {background-color:white; left:100%; top:100%;}  
    75% {background-color:black; left:50%; top:50%;}  
    100% {background-color:white; left:0%; top:0%;}  
}
```

Necessário adicionar ao <footer>  
<div id= "animation"> </div>



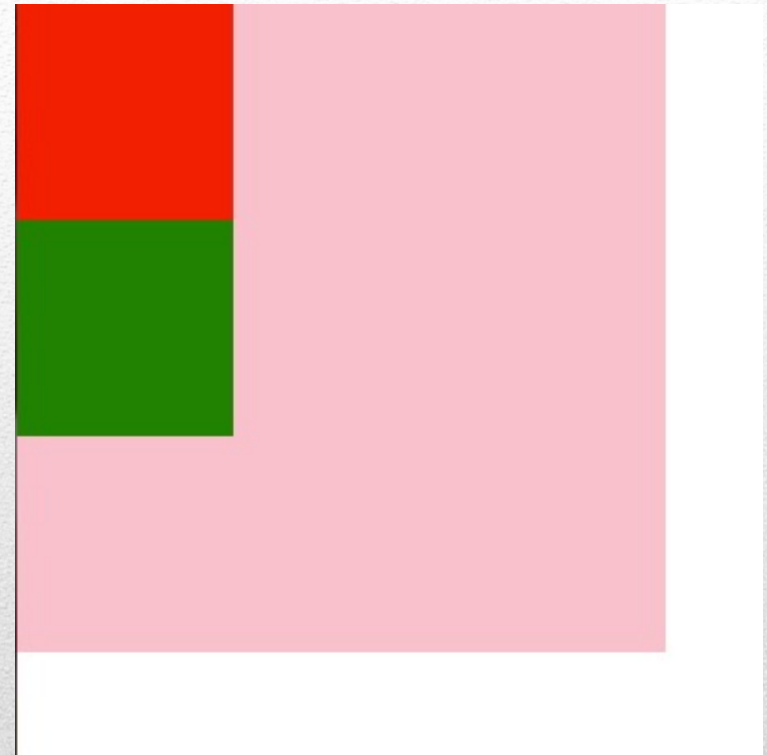
## CSS – Propriedade *Position*

### ■ Valores a atribuir à propriedade *Position*

<i>Valor</i>	Descrição
<code>static</code>	Valor por omissão. Elemento é colocado pela ordem que surge no documento html
<code>absolute</code>	Posicionamento relativo ao primeiro elemento “pai” com a propriedade “position” diferente de “static”
<code>fixed</code>	Posicionamento relativo à janela do <i>browser</i>
<code>relative</code>	Posicionamento relativo à sua posição normal, e.g., Left=20px, são adicionados 20 pixels relativamente à sua posição normal
<code>initial</code>	Volta ao valor por omissão
<code>inherit</code>	Herda o valor da propriedade <i>position</i> do elemento pai

## Propriedade “Position” (I)

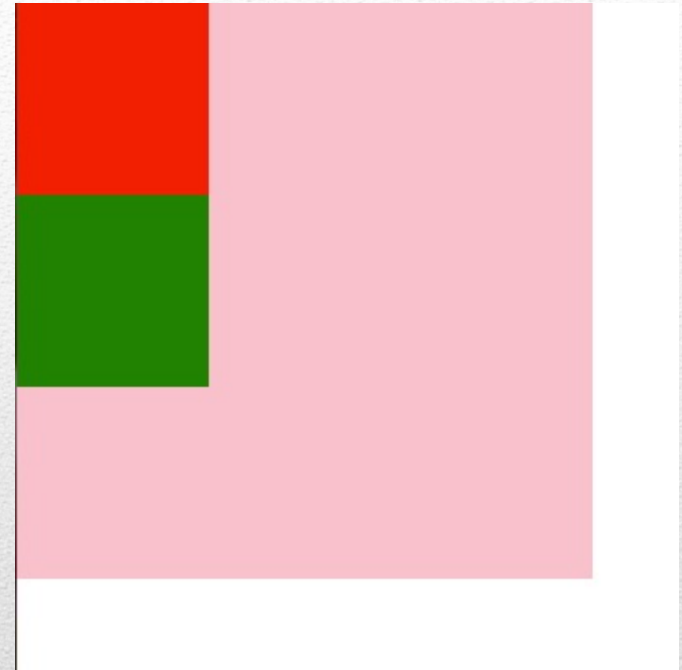
```
<body>  
  <div class="pai">  
    <div class="filho1">  
  
    </div>  
    <div class="filho2">  
  
    </div>  
  </div>  
</body>
```





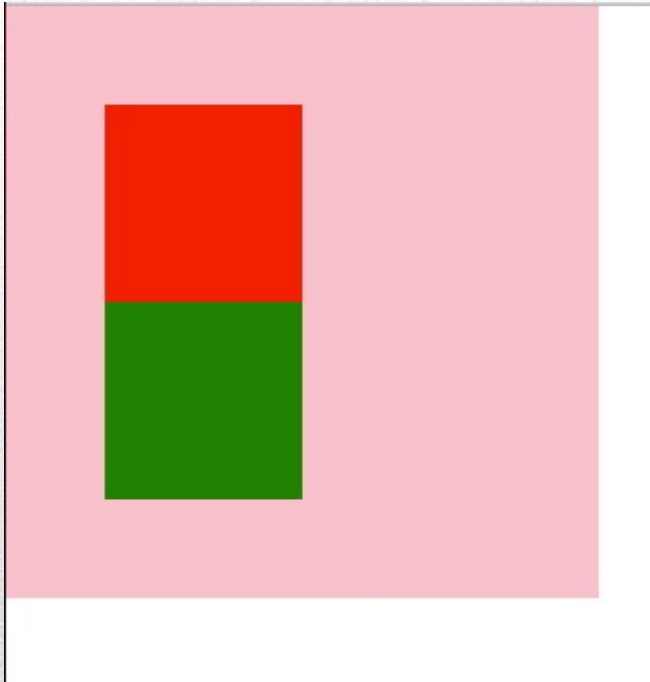
## Propriedade “Position” (II)

```
<style>
  .filho1 {
    width:100px;
    height:100px;
    background:red;
    left:50px;
    top:50px;
  }
  .filho2 {
    width:100px;
    height:100px;
    background:green;
    left:50px;
    top:50px;
  }
  .pai {
    background:pink;
    width:300px;
    height:300px;
    left: 50px;
    top:50px;
  }
</style>
```



Elementos são colocados por ordem.  
“Left” e “Top” não têm influência.

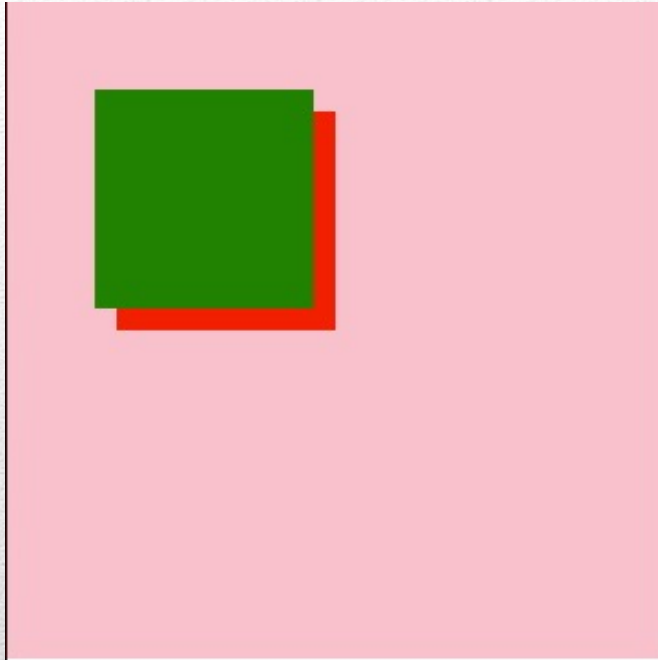
## Propriedade “Position” (III)



- **Pai** - “Position = static ”
  - **Filho1** - “Position = relative ”
  - **Filho2** - “Position = relative ”
-

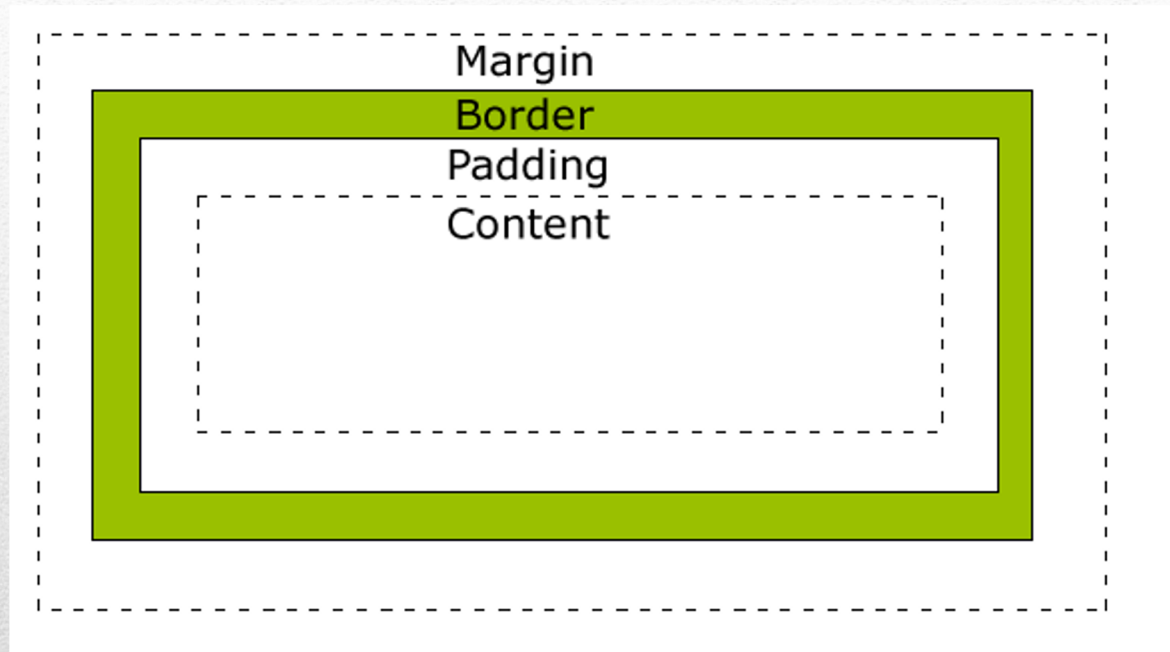


## Propriedade “Position” (IV)



- **Pai** - “Position = static ”
  - **Filho1** - “Position = absolute ”
  - **Filho2** - “Position = absolute ”
-

## CSS – Box Model



### ■ IMPORTANTE

- Atribuir valores às propriedades “**width**” e “**height**” significa definir as dimensões da área do “**content**”



## Box Model: “box-sizing”

**<h2>**Without box-sizing**</h2>**

```
<div class="div1">  
  <p> This div is smaller (width is 300px and height is 100px). </p>  
</div>  
  
<div class="div2">  
  <p> This div is bigger (width is also 300px and height is 100px). </p>  
</div>
```

**<h2>**With box-sizing (Margin not Included)**</h2>**

```
<div class="div3">  
  <p> Both divs are the same size now! </p>  
</div>  
  
<div class="div4">  
  <p>Hooray</p>  
</div>
```

---



## Box Model: Exemplo (I)

```
<style>
  .div1 {
    width: 300px;
    height: 100px;
    border-color: blue;
    border-style: solid;
  }

  .div2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    margin: 0px;
    border: 1px solid red;
  }
```

### Without box-sizing

This div is smaller (width is 300px and height is 100px).

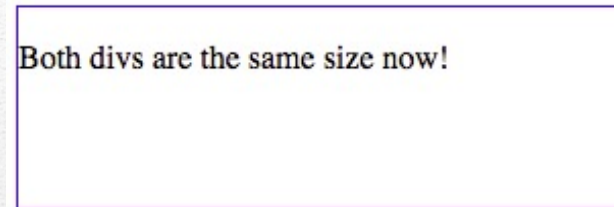
This div is bigger (width is also 300px and height is 100px).



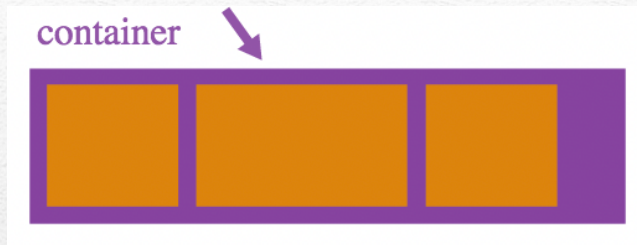
## Box Model: Exemplo (II)

```
.div3 {  
    width: 300px;  
    height: 100px;  
    margin: 0px;  
    border: 1px solid blue;  
    box-sizing: border-box;  
}  
  
.div4 {  
    width: 300px;  
    height: 100px;  
    padding: 50px;  
    border: 1px dashed red;  
    box-sizing: border-box;  
}  
</style>
```

### With box-sizing (Margin not Included)

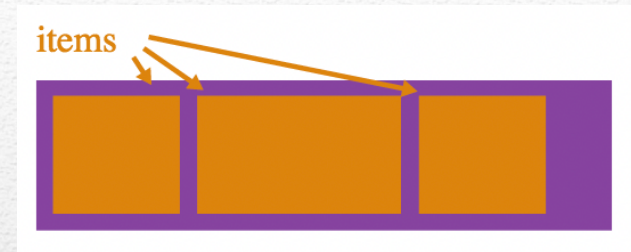


## CSS3: FlexBox



### ■ Propriedades do “container”

- **display:** flex | inline-flex
- **flex-direction:** row | column ...
- **flex-wrap:** nowrap | wrap ...
- **justify-content:** flex-start | flex-end ...
- ...

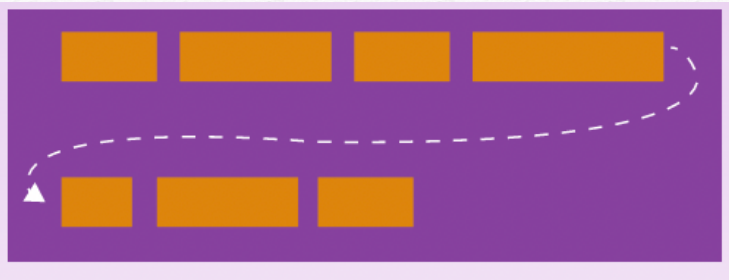


### ■ Propriedades dos “items”

- **order:** integer
- **flex:** integer
- **align-self:** flex-start | flex-end...
- ...

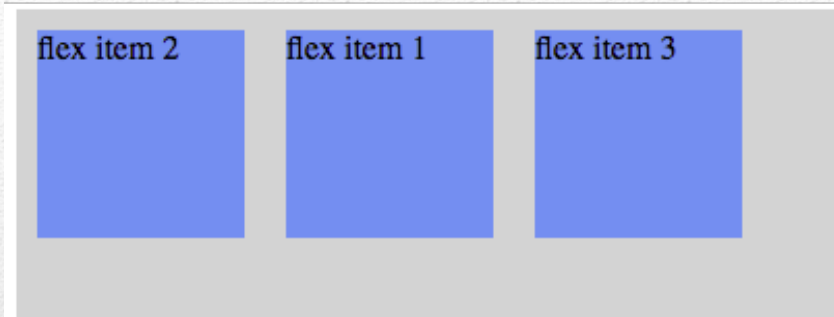


## CSS3: FlexBox – “flex-wrap”



- **flex-wrap**
  - Especifica se os items da *flexbox* vão para a linha a baixo (wrap) ou não, no caso de não haver espaço na linha

## CSS3: FlexBox – “flex-order” (I)



```
...  
<div class="flex-container">  
    <div class="flex-item">flex item 1</div>  
    <div class="flex-item first">flex item 2</div>  
    <div class="flex-item">flex item 3</div>  
</div>  
...
```

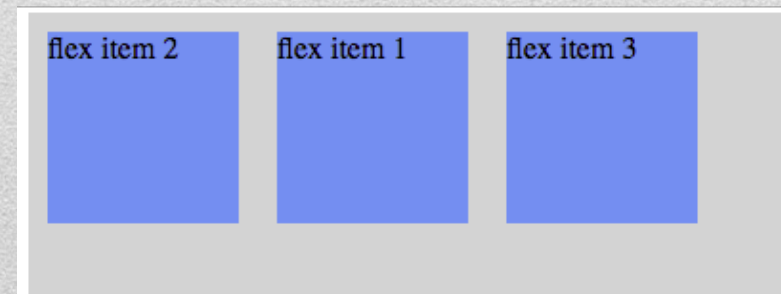


## CSS3: FlexBox – “flex-order” (II)

<style>

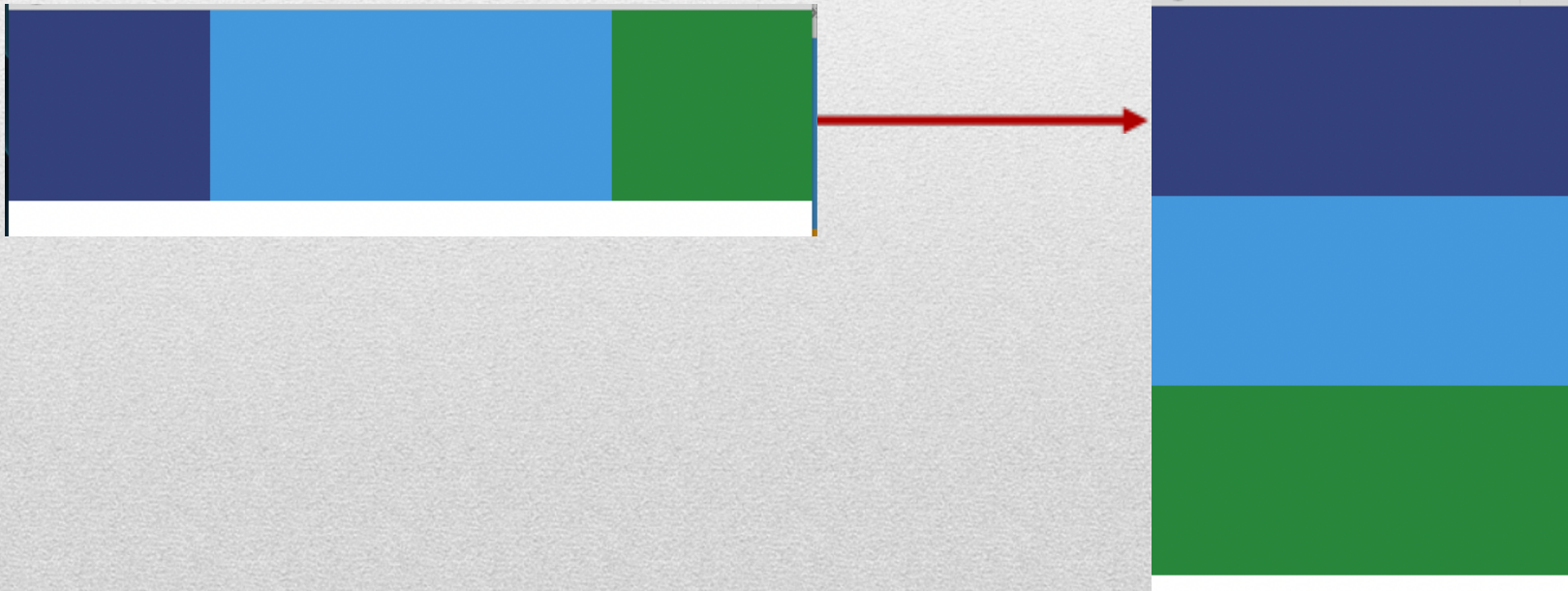
```
.flex-container {  
  display: flex;  
  width: 400px;  
  height: 250px;  
  background-color: lightgrey;  
}  
  
.flex-item {  
  background-color: cornflowerblue;  
  width: 100px;  
  height: 100px;  
  margin: 10px;  
}  
  
.first {  
  order: -1;  
}
```

</style>



## RWD: FlexBox - Exemplo (1)

```
<div class="container">  
  <div class="box dark_blue"></div>  
  <div class="box light_blue"></div>  
  <div class="box green"></div>  
</div>
```





## RWD: FlexBox - Exemplo (2)

```
.container {  
  display: flex;  
  flex-wrap: wrap;  
}  
.box {  
  width: 100%;  
}  
@media screen and (min-width: 576px) {  
  .dark_blue {  
    width: 25%;  
  }  
  .light_blue {  
    width: 75%;  
  }  
}  
@media screen and (min-width: 768px) {  
  .dark_blue, .green {  
    width: 25%;  
  }  
  .light_blue {  
    width: 50%;  
  }  
}
```

---

# Responsive Web Design(RWD) Método

---



## RWD: Introdução

- Website pode ser visualizado em vários dispositivos diferentes com qualidade idêntica



Desktop



Tablet



Smartphone

- HTML e CSS para redimensionar, esconder ou mover conteúdos
-

## RWD: Viewport

- Elemento <meta> viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

↑

Elemento <meta> “viewport” dá instruções ao *browser* como deve controlar as dimensões da página

↑

Define que a largura da página deve ser ajustada à largura do dispositivo

↑

Define o nível do *zoom* inicial quando a página é carregada

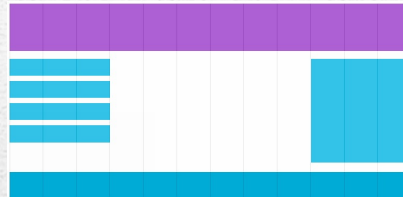
---



## RWD: 3 Conceitos Básicos

### ■ Grelhas Fluídas

- Organização do ecrã em colunas (geralmente 12) com largura proporcional às dimensões do ecrã



### ■ Media Queries

- Módulo das CSS3 que controla como os estilos são aplicados em função das características do dispositivo, por exemplo, ecrã, altura, largura ou orientação

### ■ Conteúdos Flexíveis

- Conteúdo visual que se ajuste às dimensões do ecrã. Geralmente define-se a largura como sendo um percentagem da largura da página
-

## RWD: Grelhas Fluído

- Ecrã dividido em 12 colunas

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```



- As colunas deve ter float = "left"

```
[class*="col-"] {  
    float: left;  
}
```



## RWD: Media Queries (I)

- Técnica introduzida nas CSS3
  - Utiliza a regra **@media** para incluir um bloco de propriedades das CSS apenas se uma determinada condição é verdadeira
  - Se a janela do browser tem menos de 500 pixels, a cor do fundo muda para azul claro

```
@media only screen and (max-width: 500px) {  
    body {  
        background-color: lightblue;  
    }  
}
```

---

## RWD: Media Queries - Sintaxe

- Media Queries utilizadas para verificar as capacidade dos dispositivo
  - Largura e altura do *viewport*
  - Largura e altura do dispositivo
  - Orientação do dispositivo
  - Resolução
  - ....

Não utilizando "not" ou "only" *mediatype* é opcional e todos os tipos serão implicados

- Sintaxe

```
@media not|only mediatype and (media feature) {  
    CSS-Code;  
}
```

Podemos também definir diferentes folhas de estilo para diferentes "mediatype"

```
<link rel="stylesheet" media="mediatype  
and|not|only (expressions)" href="print.css">
```



## RWD: Media Queries - mediatype

### ■ “Mediatype”

<i>Valor</i>	Descrição
<b>all</b>	Todo o tipo de media
<b>print</b>	Para impressoras
<b>screen</b>	Para ecrãs de computadores, tablets, smartphones, etc.
<b>speech</b>	Para leitores de ecrãs em voz alta

## RWD: Media Queries – media feature

- “Media Feature”

<i>Valor</i>	Descrição
<b>aspect-ratio</b>	rácio entre a largura e altura do “viewport”
<b>device-height</b>	altura do dispositivo (e.g., ecrã do PC)
<b>screen</b>	Para ecrãs de computadores, tablets, smartphones, etc.
<b>max-width</b>	máxima largura da janela do browser
<b>max-resolution</b>	máxima resolução do dispositivo em dpi's
<b>orientation</b>	a orientação do “viewport” (landscape ou portrait)



## RWD: Responsive *Breakpoints*

```
/* Small devices (landscape phones, 576px and up)*/  
@media screen and (min-width: 576px) { }
```

```
/* Medium devices (tablets, 768px and up)*/  
@media screen and (min-width: 768px) { }
```

```
/* Large devices (desktops, 992px and up)*/  
@media screen and (min-width: 992px) { }
```

```
/* Extra large devices (large desktops, 1200px and up)*/  
@media screen and (min-width: 1200px) { }
```

---

## RWD: Conteúdos Flexíveis

### ■ Imagens

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Imagem com dimensões flexíveis.  
Imagem será escalada para  
dimensões mais pequenas mas  
nunca para dimensões maiores  
que o original

```
img {  
  width: 100%;  
  height: auto;  
}
```

Imagem com dimensões  
flexíveis. Imagem será escalada  
para dimensões mais pequenas e  
maiores que o original



## RWD: Imagens Responsivas

- Imagens diferentes para diferentes dispositivos

```
/* For width smaller than 400px */  
body {  
  background-image: url('img_smallflower.jpg');  
}
```

```
/* For width 400px and larger: */  
@media only screen and (min-width: 400px) {  
  body {  
    background-image: url('img_flowers.jpg');  
  }  
}
```

---

## Boas Práticas

- [http://www.w3schools.com/html/html5\\_syntax.asp](http://www.w3schools.com/html/html5_syntax.asp)
  - <https://github.com/hail2u/html-best-practices>
  - [https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Writing\\_efficient\\_CSS](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Writing_efficient_CSS)
  - **Exemplo da Aula**
    - [https://ruimfjesus.github.io/responsive\\_website](https://ruimfjesus.github.io/responsive_website)
-